THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 6, July–August 1974
Printed in U.S.A.

Spectrum of a Binary Signal Block Coded for DC Suppression

By L. J. GREENSTEIN

(Manuscript received January 8, 1974)

This paper analyzes a block-coding scheme designed to suppress spectral energy near $f = 0$ for any binary message sequence. In this scheme, the polarity of each block is either maintained or reversed, depending on which decision drives the accumulated digit sum toward zero. The polarity of the block's last digit informs the decoder as to which decision was made.

Our objective is to derive the average power spectrum of the coded signal when the message is a random sequence of $+1$'s and -1's and the block length (M) is odd. The derivation uses a mixture of theoretical analysis and computer simulation. The theoretical analysis leads to a spectrum description in terms of a set of correlation coefficients, $\{\rho_q\}$, $q = 1, 2$, etc., with the ρ_q's functions of M. The computer simulation uses FFT algorithms to estimate the power spectrum and autocorrelation function of the block-coded signal. From these results, $\{\rho_q\}$ is estimated for various M. A mathematical approximation to ρ_q in terms of q and M is then found which permits a closed-form evaluation of the power spectrum. Comparisons between the final formula and simulation results indicate an accuracy of ± 5 percent (± 0.2 dB) or better.

The block-coding scheme treated here is of particular interest because of its practical simplicity and relative efficiency. The methods used to analyze it can be applied to other block-coding schemes as well, some of which are discussed here for purposes of comparison.

I. INTRODUCTION

1.1 Block coding

In its most general meaning, block coding consists of dividing a digital sequence into time-contiguous blocks and performing a separate coding operation on each block. In actual usage, the term is

most often applied to cases in which both the original and block-coded sequences are binary and the coding serves either (*i*) to enable error detection and/or correction or (*ii*) to shape the digital sequence spectrum.

The most widely used spectrum-shaping codes are those that suppress the energy near zero frequency. This suppression enables the use of transformers and ac-coupled amplifiers in processing the digital signal and, in modulation applications, provides for a null region near the carrier frequency to facilitate carrier extraction. Our concern here is with this kind of block coding.

In particular, we examine a block-coding approach invented (in analog form) by F. K. Bowers.[1] The digital version of this scheme has been treated separately by Carter[2] and Pierce,[3] and implemented recently by Ruthroff and Bodtmann.[4,5] The scheme can be used with blocks of either odd or even length (M), but our attention here is confined to odd M. Our objective is to derive the average power spectrum of a sequence so coded when the original message sequence is totally random. This problem has been partially studied by Rice[6] for the same block coding with M even (to which case our method of analysis is equally applicable), and by Slepian[7] and Franklin and Pierce[8] for other dc-suppressing block codes. Results for some of these cases are given later.

1.2 Description

Unless otherwise specified, the term block coding means the process we describe here, with the aid of Figs. 1 and 2.

The original sequence of binary digits is divided into blocks of length $(M - 1)$, and a $+1$ digit (the so-called code digit) is added to the end of each block (Fig. 1a). A resettable counter measures the digit sum A_k in each block k, omitting the code digit if M is even and including that digit if M is odd. In either case, this count can take on only odd values. It is compared with the sum over all previous output digits, B_k, and a decision is made as follows: If A_k and B_k have the same polarity, all pulses in block k are inverted; if A_k and B_k have opposite polarity, the pulses in block k are unaltered; and if $B_k = 0$, its polarity is taken to be that of its most recent nonzero value, i.e., B_{k-1}, or B_{k-2}, etc.* In the decoder, each received block is inverted if the polarity of

* There are other ways to resolve the case $B_k = 0$ (e.g., by random decisions, as suggested by Rice), and the ultimate choice should be dictated by practical considerations.

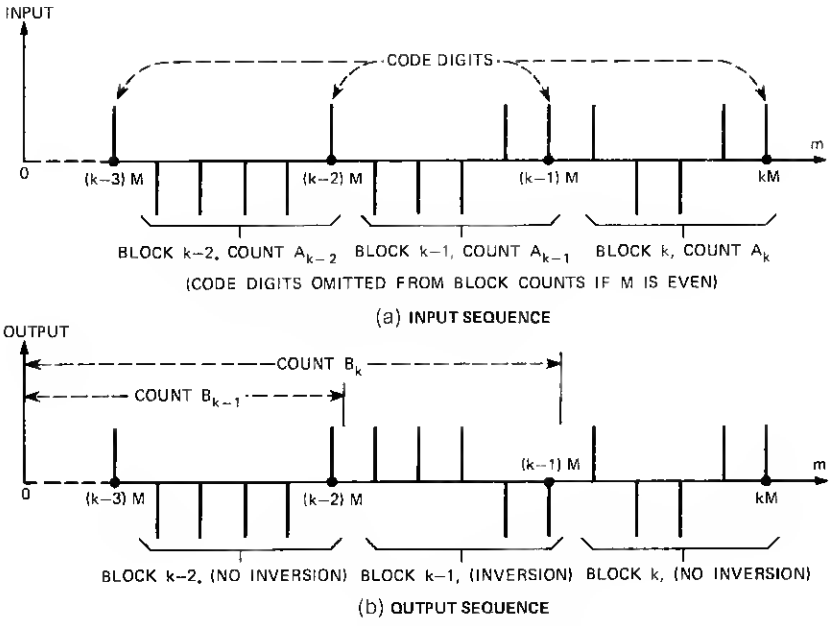

Fig. 1.—Sample input and output digital sequences.

the recovered code digit is negative and is not inverted if that digit is positive.

Figure 2 depicts the logical process just described and is a simplified diagram of how block coding is actually implemented. The identification of the code digit in the decoder is accomplished with the help of framing, which is not depicted (or treated) here. The penalties in this form of block coding are a $100/M$-percent reduction in information rate and twofold increase in the random error rate.

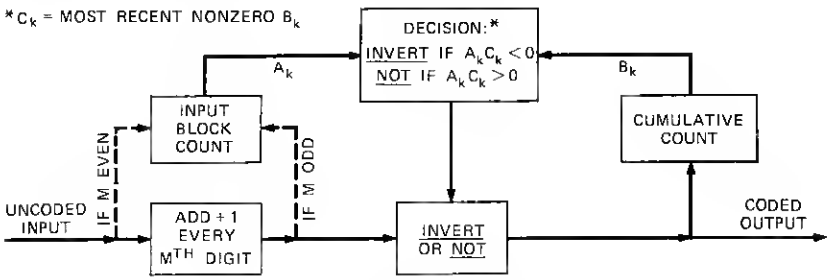

Fig. 2—Coding process.

1.3 Scope of the paper

It is easy to see that the accumulated sum of the output digits at the end of any block is limited in magnitude to M (i.e., $|B_k| \leqq M$), and that the magnitude of this sum at *any* instant in time is limited by $(3M - 1)/2$ if M is odd and $3M/2$ if M is even. For this reason, the output sequence has no spectral energy—either discrete or continuous—at $f = 0$. At the same time, the total sequence "power" is unchanged by the coding, since every digit has the same "energy," regardless of polarity.* Obviously, then, the suppressed energy near $f = 0$ is redistributed over the rest of the frequency range, and it is of more than passing interest to know how.

The answer, of course, depends on the nature of the message sequence being encoded. In this study, we assume a totally random sequence (all digits independent, with equally likely polarities) and derive the block-coded signal spectrum for odd values of M. The derivation uses a mixture of theoretical analysis and computer simulation and leads to a closed-form expression for the spectrum which compares quite favorably with simulation data. Section II gives the purely theoretical part of the derivation, Section III describes the simulation study, and Section IV gives the final result and some examples.

II. ANALYSIS

2.1 General form of the spectrum

We represent the uncoded message sequence as a binary stream of pulses at a rate $1/T$,

$$s_i(t) = \sum_{n=-\infty}^{\infty} a_n p(t - nT), \tag{1}$$

where

$$a_n = +1 \text{ or } -1 \text{ with equal probabilities,}$$

$$\overline{a_n a_m} = \begin{cases} 1 \text{ if } n = m \\ 0 \text{ if } n \neq m, \end{cases} \tag{2}$$

and $p(t)$ is a pulse centered on $[0, T]$ of arbitrary shape, area T, and Fourier transform $P(f)$. The first step in the coding consists of opening up a one-pulse slot after every $M - 1$ message pulses and injecting a positive pulse, $+p(t)$. The new sequence, with positive code pulses

* In the ensuing analysis, the digital sequence of Fig. 1 is replaced by a pulse stream at a rate $1/T$, with each pulse having an area of magnitude T.

every M time slots, is then block coded to produce an output signal

$$s_o(t) = \sum_{n=-\infty}^{\infty} b_n p(t - nT). \tag{3}$$

It is the average power spectrum of $s_o(t)$ that we wish to evaluate.

We define the autocorrelation function of the coded signal to he

$$R(\tau) = \lim_{T_o \to \infty} \left\{ \frac{1}{2T_o} \int_{-T_o}^{T_o} s_o(t) s_o(t + \tau) dt \right\} \tag{4}$$

and the average power spectrum to he the Fourier transform of $R(\tau)$,

$$S(f) = \mathfrak{F}\{R(\tau)\} = \int_{-\infty}^{\infty} R(\tau) \exp(-j\omega\tau) d\tau. \tag{5}$$

To simplify the derivation of $S(f)$, it is convenient to express $s_o(t)$, eq. (3), as the convolution

$$s_o(t) = \underbrace{\left\{ \sum_{n=-\infty}^{\infty} b_n \delta(t - nT) \right\}}_{s_u(t;\, T)} * p(t), \tag{6}$$

where $\delta(t)$ is the unit impulse function. It is now ohvious that $S(f)$ is the product

$$S(f) = S_u(f; T) |P(f)|^2, \tag{7}$$

where $S_u(f; T)$ is the average power spectrum of $s_u(t; T)$ in (6), and $P(f)$ is the Fourier transform of $p(t)$.

We can obtain $S_u(f; T)$ by applying (4) and (5) to $s_u(t; T)$. In so doing, we make use of the fact that the convolution between two unit impulse functions separated hy mT seconds is a unit impulse function $\delta(t - mT)$. It is then easy to show that

$$S_u(f; T) = \mathfrak{F}\underbrace{\left\{ \frac{1}{T} \sum_{m=-\infty}^{\infty} \left[\lim_{N \to \infty} \frac{1}{2N} \sum_{n=-N}^{N} \overline{b_n b_{n+m}} \right] \delta(\tau - mT) \right\}}_{R_u(\tau;\, T)}, \tag{8}$$

where $\overline{b_n b_{n+m}}$ is an average over the sequence ensemble and the bracketed term is the further averaging over the time position of b_n. Because $R_u(\tau; T)$ is a sequence of uniformly spaced impulses, $S_u(f; T)$ is periodic in frequency with a repetition interval $1/T$. The shaping of this spectrum by the nonperiodic pulse spectrum function $|P(f)|^2$ leads to the overall spectral characteristic of the hlock-coded signal.

2.2 Analysis of $R_u(\tau; T)$

This analysis is aimed at simplifying $R_u(\tau; T)$, eq. (8), by finding a description for $\overline{b_n b_{n+m}}$ and its average over n. To do this, we make two important observations about the block-coded binary sequence $\{b_n\}$:

(*i*) The 2^M possible digital sequences within each code block of $\{b_n\}$ are equiprobable, even though the last digit is a code digit.

(*ii*) The correlation between b_n and b_{n+m} depends only on the number of blocks separating these two digits, i.e., on the number (q) of code digits in the interval $[n, n + m)$.

The first observation is easily proved: The first $M - 1$ digits of each block at the coder input, which are assumed to be totally random, form one of 2^{M-1} equiprobable sequences. With the addition of the $+1$ code digit, there are still only 2^{M-1} realizable sequences per block. The possible inversion of the block by the coder, however, produces another 2^{M-1} realizable sequences (the original 2^{M-1} sequences with -1 instead of $+1$ for the last digit), leading to a total of 2^M. Further, since the probability of a block inversion is $\frac{1}{2}$ for a random input sequence, the 2^M realizable output sequences are equiprobable. The significance of this is that $\{b_n\}$ is statistically the same as if all M digits in each input block were derived by random selection.

The second observation depends on the first. For in the absence of block inversions and with all input digits randomly derived, there would be no correlation between any two digits of the digital stream. Any correlations in the block-coded sequence, therefore, are due solely to the inversions. It follows that $\overline{b_n b_{n+m}}$ depends, at most, on the number of possible block inversions (or code digits, q) between b_n and b_{n+m}.

We conclude that $R_u(\tau; T)$ can be expressed in terms of a set of numbers $\{\rho_q\}$, ρ_q being the correlation between any two digits having q code digits between them.* To reduce (8) to such a representation, we first observe that, if $|m| = lM + p$, where $1 \leqq p \leqq M$, then q is either l or $l + 1$, depending on the position of b_n within the block containing it. By letting n vary from the first to the last block position, we can see that $q = l$ for a fraction $[(M - p)/M]$ of all possible positions, and $q = l + 1$ for a fraction (p/M) of all possible positions. We can therefore express the bracketed quantity in (8) as

$$\left[\lim_{N\to\infty} \frac{1}{2N} \sum_{n=-N}^{N} \overline{b_n b_{n+m}}\right] = \left(\frac{M-p}{M}\right)\rho_l + \left(\frac{p}{M}\right)\rho_{l+1}, \qquad (9)$$

* It is obvious that $\rho_0 = 0$, because any two digits in the same input block are taken to be uncorrelated, and this fact is not altered by the coding.

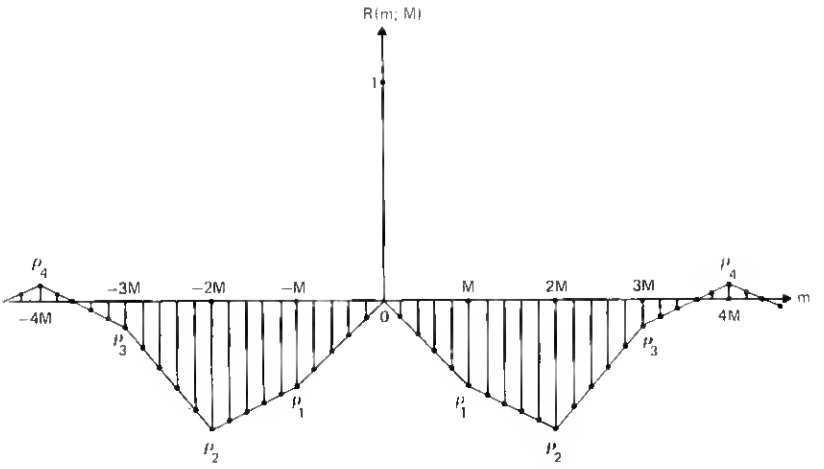

Fig. 3—Representation of $R(m; M)$.

where

$$l \triangleq \text{greatest integer in } \frac{|m|}{M}. \tag{10}$$

Since $p = |m| - lM$, the right side of (9) can be re-expressed as

$$R(m; M) \equiv \frac{(l+1)M - |m|}{M} \rho_l + \frac{|m| - lM}{M} \rho_{l+1}, \tag{11}$$

with l related to m and M by (10).

It is clear from the statistical symmetry of the b's in (9) that $R(-m; M) = R(m; M)$, and (11) reflects this fact. It is also clear from (9) that $R(0; M) = 1$. We can thus express $R_u(\tau; T)$ in (8) as

$$R_u(\tau; T) = \frac{1}{T}\left[\delta(\tau) + \sum_{m=1}^{\infty} R(m; M)\{\delta(\tau + mT) + \delta(\tau - mT)\}\right]. \tag{12}$$

Although the mathematical description for $R(m; M)$, eq. (11), seems complicated, it has the very simple graphical interpretation shown in Fig. 3. The value of $R(m; M)$ for q complete block separations (i.e., $m = \pm qM$) is just ρ_q,* and the variation between $m = qM$ and $m = (q+1)M$ is a linear progression from ρ_q to ρ_{q+1}. This result can now be used to derive $S_u(f; T)$.

* The one exception to this is the singular case $q = 0$, where $R(m = 0; M) = \rho_0 + 1$, with $\rho_0 = 0$.

2.3 Expression for $S_u(f; T)$

If the lines in Fig. 3 are envisioned as impulses of area $1/T$ separated hy T seconds, then $S_u(f; T)$ can be found as the Fourier transform of this sequence. The algehra is straightforward but somewhat tedious, and so we give just the final result:

$$S_u(f; T) = \frac{1}{T}\left[1 + 2M\left(\frac{\sin M\omega T/2}{M \sin \omega T/2}\right)^2 \sum_{q=1}^{\infty} \rho_q \cos \omega MT\right]. \quad (13)$$

The periodicity of this function, with repetition interval $1/T$, is easy to see. The ρ_q's are functions of M so that a complete description of $S_u(f; T)$, and thus of $S(f)$ as given by (7), reduces to knowing the array of functions $\{\rho_q(M)\}$. Unfortunately, there is no apparent way to determine these functions from purely theorctical considerations. One useful bit of information, however, is that $S_u(0; T) = 0$ hy virtue of the block coding.* This heing the case, we see from (13) that

$$\sum_{q=1}^{\infty} \rho_q(M) = -\frac{1}{2M}. \quad (14)$$

Beyond (7), (13), and (14), we have little information ahout the block-coded signal spectrum on theoretical grounds. Using the methods of computer simulation, however, it is possible to estimate the ρ_q's for various M, and to seek functional descriptions for them that permit a closed-form evaluation of (13). This task constitutes the remainder of the development.

III. SIMULATION STUDY

3.1 Computer programs

The computer programs used to derive $\{\rho_q(M)\}$ empirically are depicted in Fig. 4. The routine called BLOCK generates random sequences $\{a_n\}$ having the properties described hy (2) and, for specified M, converts them into hlock-coded sequences $\{b_n\}$ by emulating the logic in Fig. 2. These coded sequences are supplied on demand to the main program, labelled SIMULATION.

The SIMULATION program operates in the following manner: In each of N_T trials, it accepts an N-term sequence from BLOCK and performs an N-point discrete Fourier transform (DFT[9]), producing complex spectral samples at $f = k/NT$, $k = 0, N - 1$. The squared magnitude of the kth sample (normalized by N) represents a one-trial

* This is so because the long-time integration of the coded sequence is bounded in magnitude (specifically, by $3M/2$).

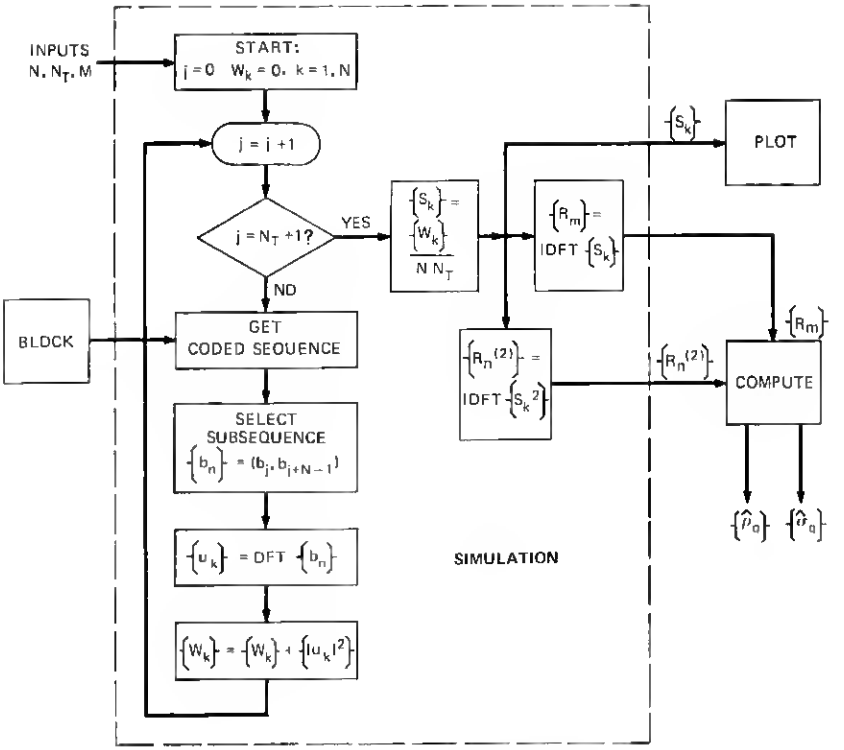

Fig. 4—Computer simulation program.

estimate of $TS_u(f = k/NT; T)$. These estimates for the N_T trials are averaged to produce the array $\{S_k\}$, $k = 0, N - 1$. This procedure is made efficient by implementing the DFT's with fast Fourier transform (FFT) algorithms.[10] To maximize efficiency, N is constrained in all simulations to be an integral power of 2.

The full benefit of the multitrial averaging is obtained by enforcing statistical independence among the N_T sequences supplied by BLOCK, and also by effectively randomizing the time phase of the sequences analyzed. The latter is accomplished by means of the function labelled "SELECT SUBSEQUENCE . . ." (Fig. 4), which causes the starting time of the analyzed sequences to vary uniformly among the M possible positions within a block. To accommodate this feature, the independent sequences supplied by BLOCK have a total length $N + N_T - 1$ or greater.

There are three output arrays produced in the SIMULATION program. One is $\{S_k\}$, which approximates $TS_u(f; T)$ at the N frequencies

k/NT, $k = 0, N - 1$. This array is applied to the PLOT routine, Fig. 4, where it is plotted on the same graph as the mathematical function given by (28). This function is derived using numerical parameters extracted from $\{R_m\}$, the inverse DFT of $\{S_k\}$, which is the second output array, Fig. 4. The elements of this array represent estimates of $R(m; M)$, i.e., $R_m \doteq R(m; M)$, and they are applied to the COMPUTE routine to produce estimates of ρ_q, i.e., the array $\{\hat{\rho}_q\}$. These are the quantities used to derive (28) from (13). To evaluate the accuracy of these estimates, the COMPUTE routine also produces the array $\{\hat{\sigma}_q\}$, where $\hat{\sigma}_q$ is the approximate rms error in $\hat{\rho}_q$. The computation of $\{\hat{\sigma}_q\}$ involves the array $\{R_n^{(2)}\}$, the inverse DFT of $\{S_k^2\}$, which is the third output of the SIMULATION program. The formulas relating $\{R_m\}$, $\{\hat{\rho}_q\}$, $\{R_n^{(2)}\}$, and $\{\hat{\sigma}_q\}$ are presented in Section 3.3.

3.2 Choices of N and N_T

The difference between the computer-derived spectral sample S_k and the quantity it approximates, $TS_u(k/NT; T)$, contains two distinct components, (*i*) a deterministic error due to the finiteness (N) of the sample length and (*ii*) a random error due to the finiteness (N_T) of the number of independent simulations. Similar remarks apply to the difference between R_m and $R(m; M)$. We now apply these considerations to the choices of N and N_T.

Because N is finite, the normalized spectrum estimated by the computer program is not $TS_u(f; T)$, but the convolution between $TS_u(f; T)$ and the function

$$F(f) = NT\left(\frac{\sin \pi N f T}{\pi N f T}\right)^2. \tag{15}$$

Thus, S_k is an estimate of the quantity

$$TS_u'\left(\frac{k}{NT}; T\right) = \int_{-\infty}^{\infty} TS_u(f; T) F\left(f - \frac{k}{NT}\right) df. \tag{16}$$

Since the area of $F(f)$ is unity, the approximation of $TS_u(k/NT; T)$ by $TS_u'(k/NT; T)$ is very good if $TS_u(f; T)$ changes negligibly over the main lobe of $F(f - k/NT)$. The difference is the deterministic error in S_k; the inverse DFT of the k-sequence of these errors gives the deterministic errors in the estimates of $R(m; M)$.

By considering the interference between the peak of $TS_u(f; T)$ (which occurs near $f = 1/4MT$) and the sidelobes of $F(f)$ (which decrease as $1/f^2$), we have determined a rule of thumb for which worst-case deterministic errors are negligibly small. The rule constrains N

to the region

$$N \geqq 100\, M, \tag{17}$$

a constraint we have used throughout this study.

The random errors in estimating $R(m; M)$ also decrease with increasing N, as we shall see in Section 3.3. This consideration, added to (17) and the requirement that N he an integral power of 2, has helped to shape the final choices of N for different values of M. Typically, we have used $N = 512$ for $M = 3$, $N = 2048$ for $M = 9$, and $N = 4096$ for $M = 17$.

We shall also see in Section 3.3 that the random errors in estimating $S_u(f; T)$ and $R(m; M)$ decrease with increasing N_T. For example, the fractional rms error in S_k is accurately given by $1/\sqrt{N_T}$. In deriving estimates for $R(m; M)$, we have used 400 trials to achieve the accuracies desired, while, to ohtain precise spectral estimates for comparison with the final formula, we have used 1600 trials (corresponding to ± 2.5 percent accuracy).

3.3 Analysis of computer results

The simulation estimates of $R(m; M)$ for $m > 0$ and $M = 3$, 9, and 17 are given hy the points in Figs. 5, 6, and 7. The existence of straight-line variations hetween $m = qM$ and $m = (q + 1)M$ for $q = 0$, 1, 2, etc., as predicted hy the analysis of Section II, is evidenced here. The deviations of the points from straight lines are due to statistical fluctuations in the finite simulation, and the straight lines shown are derived from the points hy least-squares techniques. The pertinent error analyses and estimation procedures used to ohtain these straight lines and further data reductions are now summarized. We assume from here on that deterministic errors are made negligihle hy the choice of N, i.e., that all errors in S_k and R_m are random errors due to finite N_T. Tahle I lists the symhols to he used.

3.3.1 Error correlations

We now estahlish the error correlations $\overline{\delta_k \delta_l}$ and $\overline{\epsilon_m \epsilon_p}$ with the aid of the definitions in Table I. As N becomes very large, the real and imaginary parts of u_k hecome more and more like independent gaussian variates (central limit theorem[11]), and we assume this to be the case here. The importance of this assumption is that the definitions of δ_k, S_k, and $\overline{S_k}$ in Tahle I can then he used to ohtain

$$\overline{\delta_k \delta_l} = \frac{\overline{|u_k u_l^*|^2}}{N^2 N_T}. \tag{18}$$

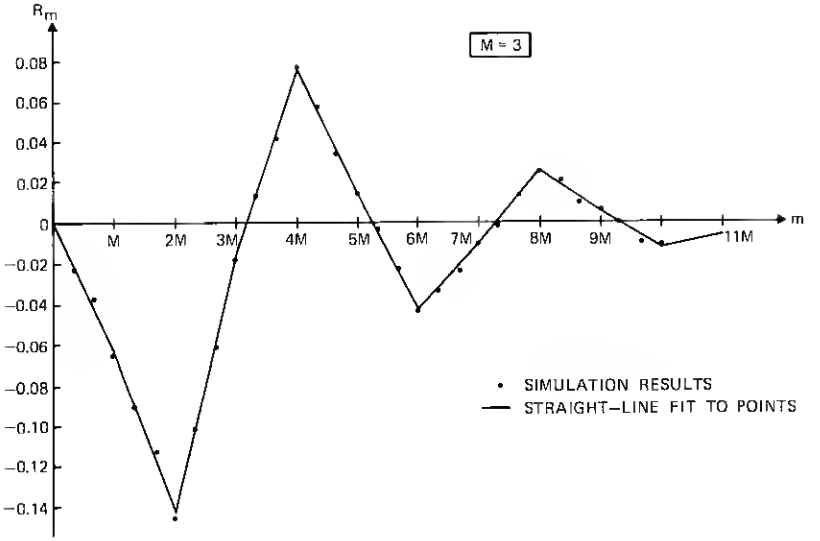

Fig. 5—Simulation results for $M = 3$ ($N = 512$, $N_T = 400$ trials).

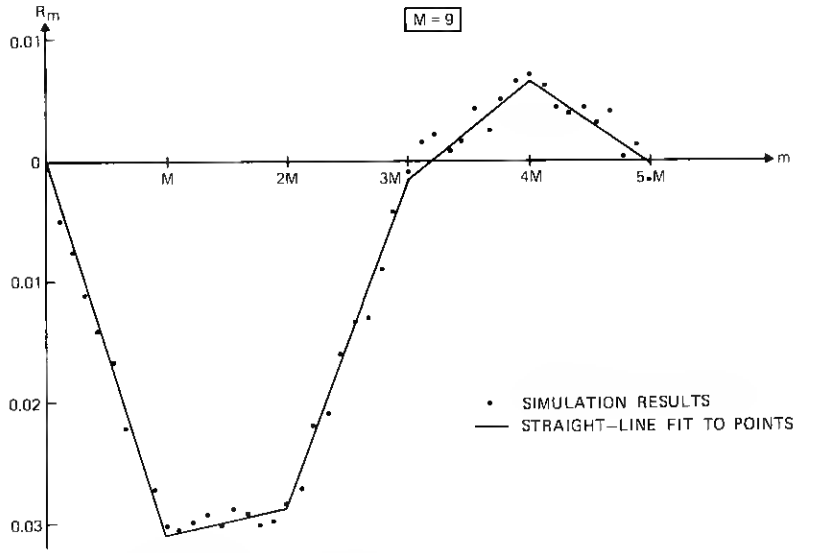

Fig. 6—Simulation results for $M = 9$ ($N = 2048$, $N_T = 400$ trials).

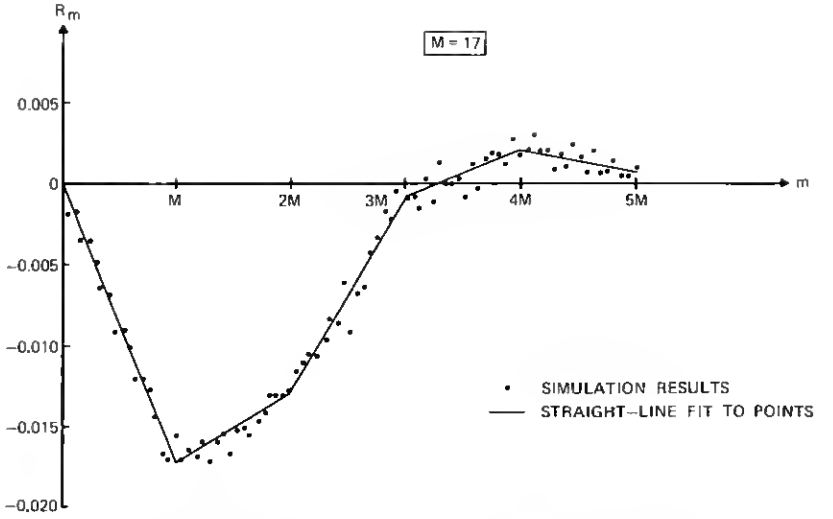

Fig. 7—Simulation results for $M = 17$ ($N = 4096$, $N_T = 400$ trials).

Clearly, $\overline{\delta_k^2} = \overline{(S_k)^2}/N_T$, i.e., the rms error in estimating $\overline{S_k}$ is $\overline{S_k}/\sqrt{N_T}$. Unfortunately, $\overline{\delta_k \delta_l}$ for $l \neq k$ is not zero for the block-coded signal under study. Nevertheless, these correlations are found to be sufficiently weak (particularly as $|k - l|$ increases) that we can ignore them without any first-order effects on the results. The benefit of this is a considerable simplification in the mathematics. Accordingly, we shall assume that

$$\overline{\delta_k \delta_l} \doteq \begin{cases} \overline{(S_k)^2}/N_T; & l = k \\ 0; & l \neq k. \end{cases} \tag{19}$$

Table I — Symbols used in error analysis

Symbol	Meaning
$\{u_k\}$, $k = 0, N-1$	DFT of $\{b_n\}$, $n = 0, N-1$
$S_u(f; T)$	Average power spectrum of infinitely long block-coded signal
S_k	Simulation estimate of $S_u(f = kT/N; T)$: $S_k = \underset{N_T}{\text{Ave}}(\lvert u_k \rvert^2/N)$
$\overline{S_k}$	Limiting value of S_k as $N_T \to \infty$: $\overline{S_k} = \overline{\lvert u_k \rvert^2}/N$
δ_k	Random error in S_k: $\delta_k = S_k - \overline{S_k}$
$R(m; M)$	Coefficient, at separation m, of autocorrelation function of block coded signal
R_m	Simulation estimate of $R(m; M)$: $\{R_m\} = \text{IDFT}\{S_k\}$
ϵ_m	Random error in R_m: $\epsilon_m = R_m - R(m; M) = \text{IDFT}\{\delta_k\}$
ρ_q, $\hat{\rho}_q$	Exact, estimated values of $R(m = qM; M)$

To establish $\overline{\epsilon_m \epsilon_p}$, we begin with the fact that $\{\epsilon_m\}$ is the inverse DFT of $\{\delta_k\}$. Using (19) and a modest amount of manipulation, we can then show that

$$\overline{\epsilon_m \epsilon_p} = \frac{2}{NN_T} \underbrace{[\text{IDFT}\{(\overline{S_k})^2\}|_{m+p}}_{R^{(2)}_{m+p}} + \underbrace{\text{IDFT}\{(\overline{S_k})^2\}|_{|m-p|}]}_{R^{(2)}_{|m-p|}}. \quad (20)$$

In the SIMULATION program (see Fig. 4), the array $\{R_n^{(2)}\}$ is estimated by computing the inverse DFT of $\{S_k^2\}$, and is then used to determine the rms errors in the estimates for $\rho_q(M)$.

3.3.2 Estimations of $\rho_q(M)$

Given the simulation estimates $\{R_m\}$, an obvious way to estimate $\rho_q (\equiv R(m = qM; M))$ is by $\hat{\rho}_q = R_{m=qM}$. However, the values of R_m for $m = qM - 1,\ qM - 2,\ \cdots qM - (M - 1)$ can be included to yield more accurate estimates of ρ_q, as the following analysis shows.

Using Table I and Fig. 3, we can express R_m in the general form

$$R_m = \rho_{q-1} + \frac{\rho_q - \rho_{q-1}}{M}[m - (q-1)M] + \epsilon_m; \quad (q-1)M < m \leqq qM$$
$$q = 1, 2, \text{ etc.} \quad (21)$$

Now suppose that we estimate ρ_q as a linearly weighted sum of R_m over $[(q-1)M + 1, qM]$. Using the substitution $m' = m + (q-1)M$, this sum can be written as

$$\hat{\rho}_q = \sum_{m=1}^{M} w_m R_{m'} = \rho_{q-1} \sum_{m=1}^{M} w_m + \frac{\rho_q - \rho_{q-1}}{M} \sum_{m=1}^{M} m w_m + \sum_{m=1}^{M} w_m \epsilon_{m'}. \quad (22)$$

This estimate is made unbiased by choosing $\{w_m\}$ so that

$$\sum_{m=1}^{M} w_m = \begin{cases} \text{Arbitrary}, & q = 1 \\ 1, & q > 1 \end{cases} \quad \text{and} \quad \sum_{m=1}^{M} m w_m = M. \quad (23)$$

(For the singular case $q = 1$, there is no constraint on $\sum w_m$ because ρ_0 is known to be zero.) To see how to choose $\{w_m\}$ $(m = 1, M)$ within these constraints, we combine (20), (22), and (23) and obtain the following mean-square error for $\hat{\rho}_q$:

$$\sigma_q^2 = \sum_{m=1}^{M} \sum_{p=1}^{M} w_m w_p \overline{\epsilon_{m'} \epsilon_{p'}} = \frac{2}{NN_T} \sum_{m=1}^{M} \sum_{p=1}^{M} w_m w_p [R^{(2)}_{m'+p'} + R^{(2)}_{|m'-p'|}]. \quad (24)$$

To a first approximation, the dominant component of σ_q^2 is $2R_0^{(2)}(\sum w_m^2)/$

Table II — Estimates of $\rho_q(M)$

q \ M	3 ($\sigma \doteq 3.1\times10^{-3}$)	5 ($\sigma \doteq 1.8\times10^{-3}$)	9 ($\sigma \doteq 1.0\times10^{-3}$)	13 ($\sigma \doteq 6.1\times10^{-4}$)	17 ($\sigma \doteq 5.3\times10^{-4}$)
1	−0.06321	−0.04787	−0.03098	−0.02271	−0.00172
2	−0.14363	−0.06625	−0.02870	−0.01751	−0.00128
3	−0.01754	−0.00632	−0.00168	−0.00140	−0.00081
4	0.07654	0.02712	0.00659	0.00398	0.00212
5	0.01431	0.00515	−0.00018	0.00024	0.00066
6	−0.04212	−0.01120	−0.00198	−0.00077	−0.00091
7	−0.01074	−0.00200	−0.00075	0.00075	−0.00041
8	0.02653	0.00701	0.0	0.00002	0.00103
9	0.00546	−0.00018	0.00069	0.00039	0.00088
10	−0.01244	−0.00483	−0.00026	0.00069	0.00020
11	−0.00600	−0.00040	−0.00016	−0.00009	−0.00050
12	0.01124	0.00185	−0.00103	0.00017	0.00061
13	0.00275	0.00143	0.00183	0.00024	−0.00031
14	−0.00471	−0.00096	0.00009	0.00032	−0.00017
15	−0.00062	0.00022	−0.00008	0.00003	0.00018
16	0.00038	0.00200	−0.00062	0.00061	0.00027
17	0.00171	0.00016	0.00013	−0.00023	0.00017
18	0.00027	−0.00407	−0.00087	−0.00050	0.00004
19	0.00254	−0.00134	0.00039	0.00047	−0.00019
20	0.00024	0.00201	0.00057	0.00051	0.00024

NN_T, because $|R_n^{(2)}|$ tends to be small for $n \neq 0$. Using this fact, an approximate least-squares approach is to derive the sequence $\{w_m\}$ for which $\sum w_m^2$ is a minimum within the constraints of (23). Using Lagrangian multipliers, it is straightforward to show that the solution is

$$w_m \underset{(m=1,M)}{=} \begin{cases} -\dfrac{2}{M} + \dfrac{6m}{M(M+1)}\,; & q > 1 \\ \dfrac{6m}{(M+1)(2M+1)}\,; & q = 1. \end{cases} \tag{25}$$

We assume that, for practical purposes, (25) represents the least-squares coefficient array for the error given by (24). It was used in the COMPUTE routine of Fig. 4 to obtain $\{\hat{\rho}_q\}$ [based on (22) and the estimates $\{R_m\}$] and to estimate $\{\sigma_q\}$ [based on (24) and the estimates of $\{R_n^{(2)}\}$].

The results are shown in Table II for several values of M and for $q = 1$, 20. It is found that σ_q is fairly constant with q, except for σ_1, which tends to be lower by 10 to 30 percent. The quantity σ in each column heading of Table II is the average of the computed σ_q's from $q = 2$ to $q = 20$.* These rms errors are lower than those obtained

* Note that, for all M, $\hat{\rho}_q$ is in the simulation "noise" (i.e., $|\hat{\rho}_q| \leqq \sigma$) for $q \geqq 20$.

by estimating ρ_q as $R_{m=qM}$ [which is equivalent to using $\{w_m\} = (0, 0, \cdots, 1)$], the improvement factor increasing with M and having a value near 2.5 for $M = 17$.

3.4 Reduction to mathematical descriptions

From the data of Table II, a useful and valid description for $\hat{\rho}_q(M)$ can be shown to be

$$\hat{\rho}_q(M) \doteq \frac{A_q}{M} + \frac{B_q}{M^2}, \tag{26}$$

where A_q and B_q are functions solely of q. For each q, raw estimates of A_q and B_q are derivable from the $\hat{\rho}_q$ values at any two values of M. To satisfy (14) for all M, however, it is necessary that these estimates be

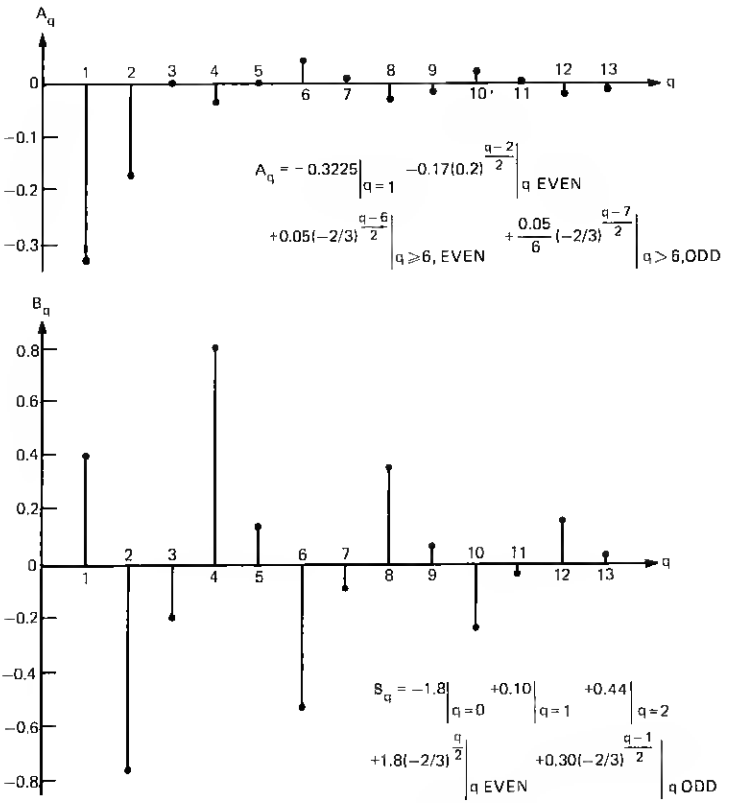

Fig. 8—Possible solutions for $\{A_q\}$ and $\{B_q\}$.

refined to satisfy

$$\sum_{q=1}^{\infty} A_q = -\frac{1}{2M}, \qquad \sum_{q=1}^{\infty} B_q = 0. \tag{27}$$

Moreover, these refined estimates should express A_q and B_q in mathematical forms that will permit a closed-form evaluation of (13). One possible solution that satisfies all these requirements is Fig. 8. Using these results in (26) and comparing with the values in Table II, agreement is found to be quite good: Nearly all the new estimates of $\hat{\rho}_q$ obtained in this way lie within a standard deviation ($\pm\sigma$) of the tabulated values; also, the values of $\hat{\rho}_q$ for $q > 20$ lie within $\pm\sigma$ about zero, decaying in magnitude with q as expected from physical reasoning.

IV. FINAL RESULT AND EXAMPLES

4.1 Expression for $S_u(f; T)$

Combining Fig. 8 with (26) and (13), it is possible to obtain a closed-form expression for $S_u(f; T)$. Once again, the algebra is tedious but straigbtforward, so we merely state the result:

$$TS_u(f; T) = 1 - \left(\frac{\sin(M\omega T/2)}{M \sin(\omega T/2)}\right)^2 F(\omega T; M), \tag{28a}$$

where

$$\begin{aligned} F(\omega T; M) &= 0.645 \cos(M\omega T) - \frac{0.88}{M} \cos(2M\omega T) \\ &\quad - 0.85 \frac{\cos(2M\omega T) - 0.2}{\cos(2M\omega T) - 2.6} - \frac{1}{13 + 12\cos(2M\omega T)} \\ &\quad \times \left[\frac{2}{M}[-7.2 + 6.4\cos(M\omega T) - 10.8\cos(2M\omega T) \right. \\ &\quad + 0.6\cos(3M\omega T)] + 0.05[12\cos(4M\omega T) + 2\cos(5M\omega T) \\ &\quad \left. + 18\cos(6M\omega T) + 3\cos(7M\omega T)]\right]; \quad M \text{ odd.} \end{aligned} \tag{28b}$$

Rather than do an error analysis of this result (e.g., based on the σ's in Table II), we have compared this formula with fresh simulation results for $\{S_k\}$ based on 1600 trials (± 2.5 percent rms error). The PLOT routine shown in Fig. 4 plots the simulation data as points and plots the formula as a solid line. Figures 9 through 13 give the results for $M = 3$, 5, 9, 13, and 17. Given the rms errors of the simulations and the scatter about the solid curves, we estimate from these comparisons that the formula is accurate to within ± 5 percent (± 0.2 dB) or better for all M and ω. The accuracy is especially good in the all-

important rising portions near $f = 0$, where the simulation points are seen to lie very close to—or within the line thickness of—the solid curves.

4.2 Comparison with even-M block code

It is tempting to extrapolate the new formula to the case of even M, although Fig. 2 warns us that the coding schemes for odd and even M are qualitatively different. Figure 14 shows simulation points, along with a solid curve derived from the new formula, for $M = 4$. Figures 15 and 16 do the same for $M = 8$ and 16. Although N and N_T are lower in these simulations, the consequent increases in the deterministic and random errors do not account for the observed discrepancies. It is concluded that the new formula is not accurate for low even values of M, but that its accuracy improves as M increases to large even values.

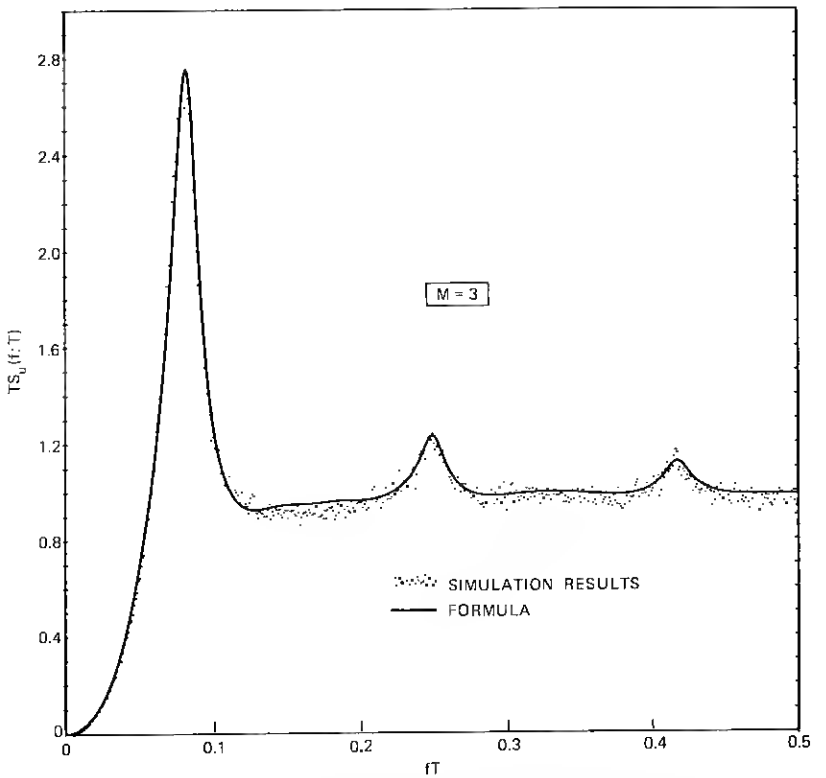

Fig. 9—Comparison of formula with simulation results for $M = 3$ ($N_T = 1600$).

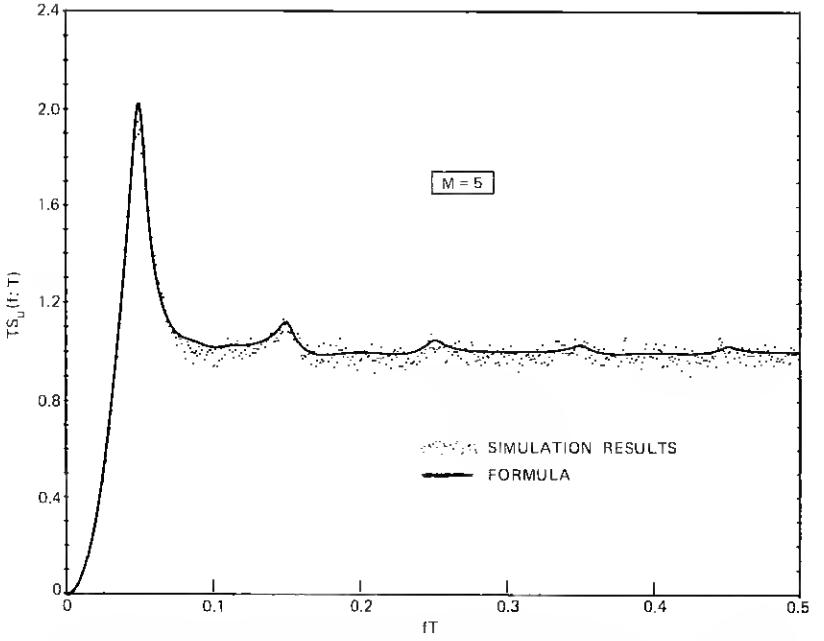

Fig. 10—Comparison of formula with simulation results for $M = 5$ ($N_T = 1600$).

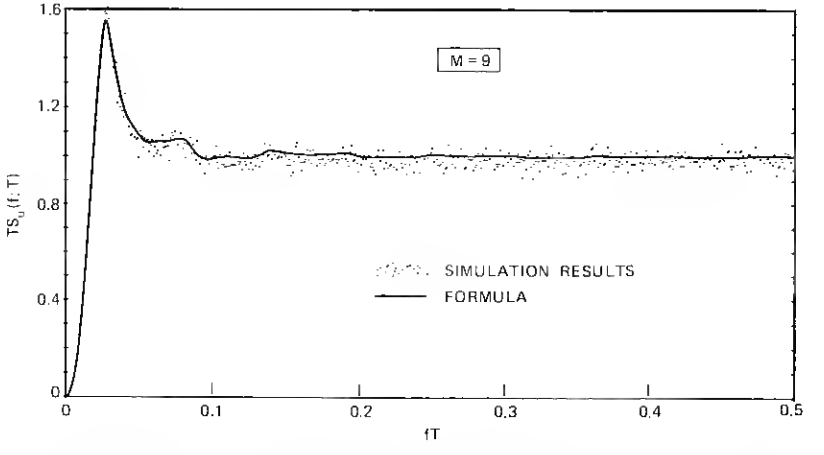

Fig. 11—Comparison of formula with simulation results for $M = 9$ ($N_T = 1600$).

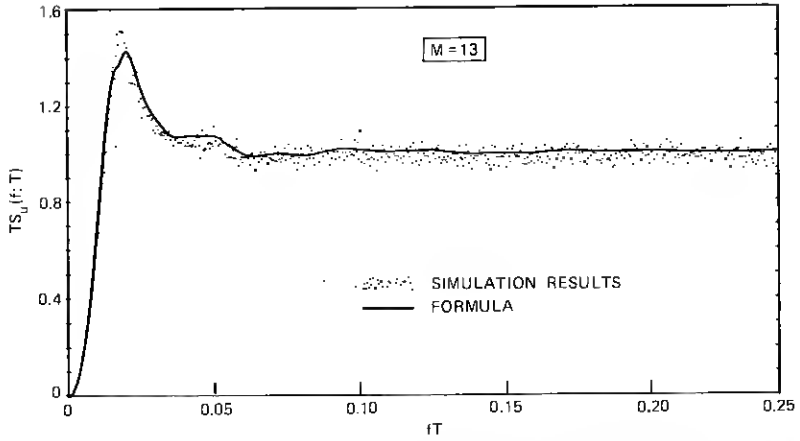

Fig. 12—Comparison of formula with simulation results for $M = 13$ ($N_T = 1600$).

4.3 Comparison with zero-disparity code

A different kind of block-coding scheme is one in which M is even and each block is constrained to have an equal number of positive and negative digits.[2] For a given M, this so-called zero-disparity code is less efficient in information rate than the one studied here (see Franklin and Pierce[8]), but has superior spectral properties, as we now show.

For the zero-disparity code, Franklin and Pierce show that $TS_u(f; T)$ is $M/(M - 1)$ times the function (28a), with $F(\omega T; M)$ replaced by 1.

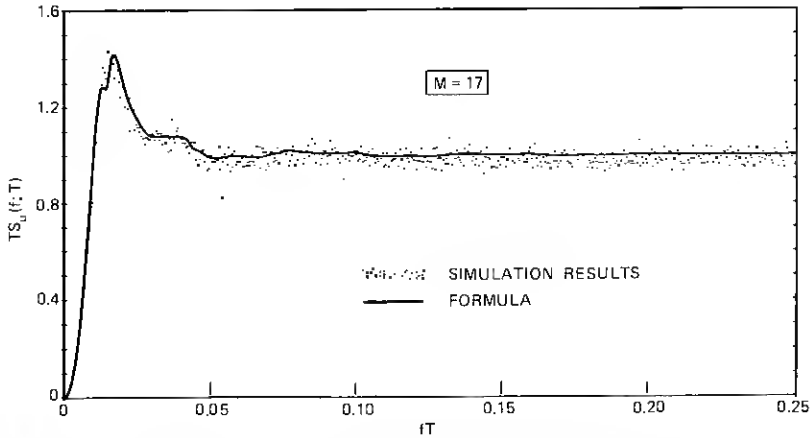

Fig. 13—Comparison of formula with simulation results for $M = 17$ ($N_T = 1600$).

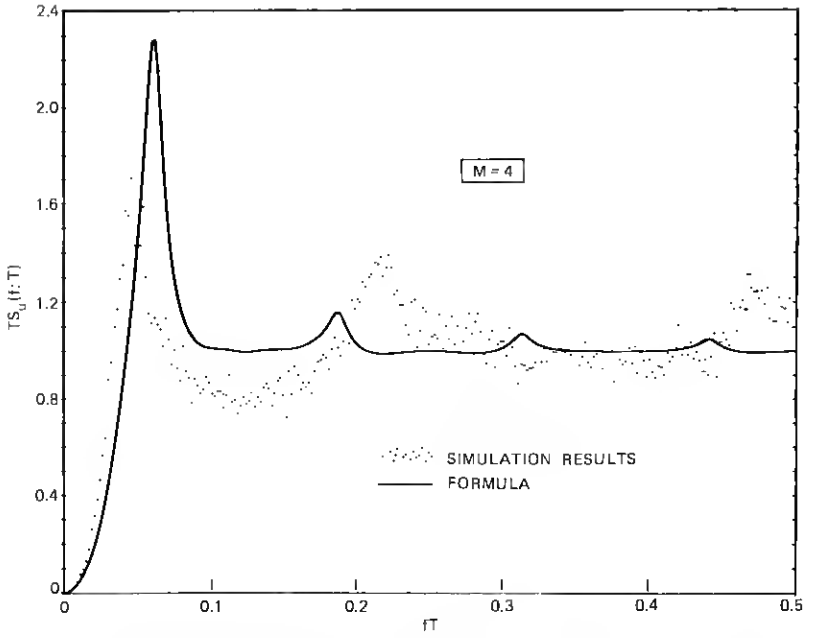

Fig. 14—Comparison of formula with simulation results for $M = 4$ ($N_T = 400$).

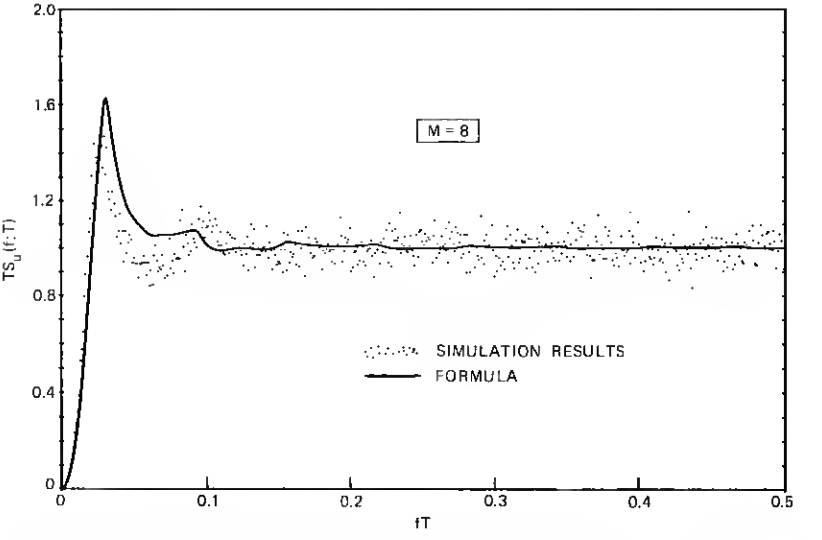

Fig. 15—Comparison of formula with simulation results for $M = 8$ ($N_T = 400$).

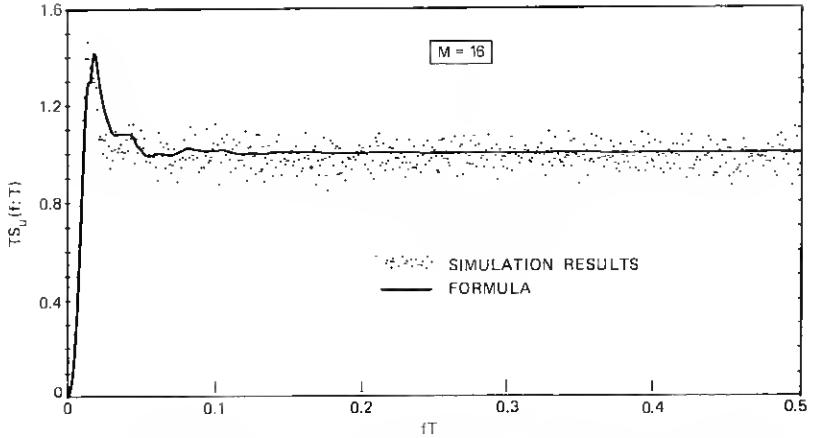

Fig. 16—Comparison of formula with simulation results for $M = 16$ ($N_T = 400$).

For $f \leqq 1/2T$ and $M \geqq 4$, this result can be represented to within 0.3 dB as follows:

$$\frac{M-1}{M}\, TS_u(f;T) \doteq 1 - \left(\frac{\sin\,(M\omega T/2)}{M\omega T/2}\right)^2. \tag{29}$$

This function is shown (dashed curve) in Fig. 17 and compared with

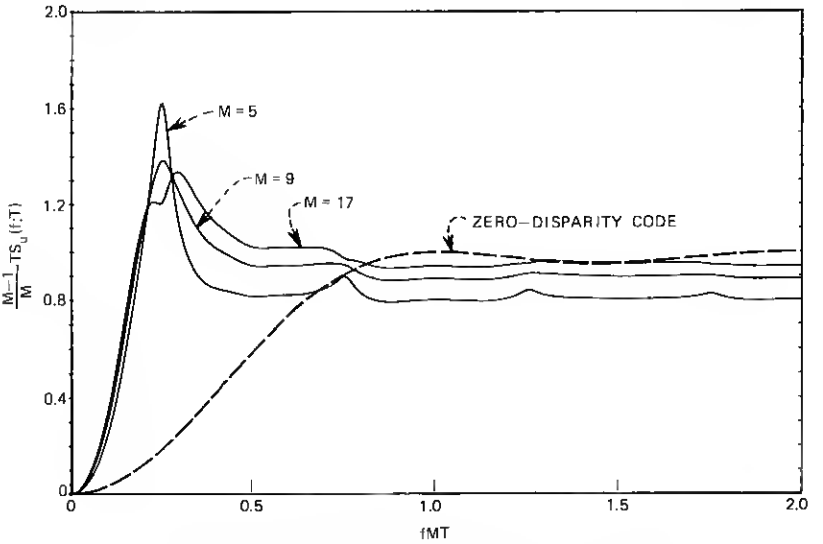

Fig. 17—Comparisons with zero-disparity block code.

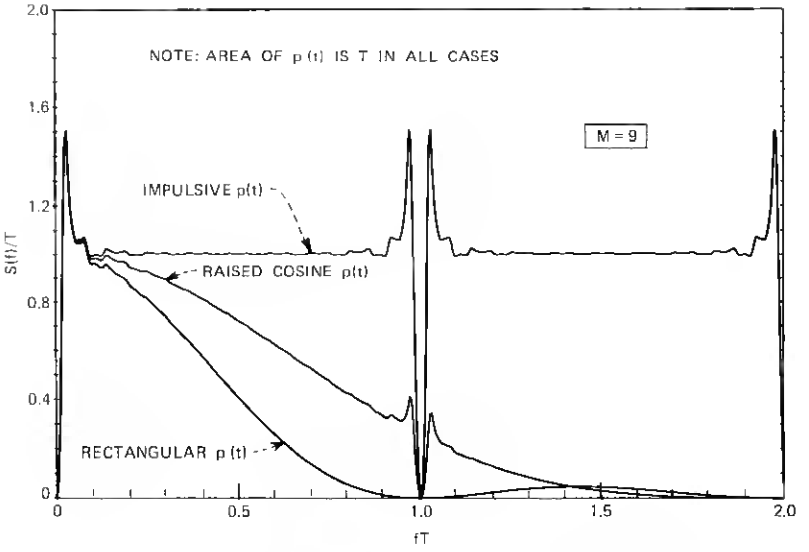

Fig. 18—Block-coded signal spectrum for various pulse shapes ($M = 9$).

some corresponding results for the block code studied here. The latter results, based on the new formula, are for $M = 5$, 9, and 17. The suppressed energy near $f = 0$ is seen to be redistributed to higher frequencies (and in a more uniform way) by the zero-disparity code, permitting more relaxed requirements on ac-coupled processing stages for a given M. If the quantity held fixed is information rate, however, the zero-disparity code must use a larger value of M for which its spectral superiority all but vanishes.

4.4 Effects of pulse shape

The overall spectrum of the block-coded signal must take into account the spectrum of the pulse shape $p(t)$. Figure 18 gives some results for $S(f)$, (7), for the case $M = 9$. The relative differences due to pulse shape are identical to those that occur without block coding. The effect of the block coding is to force spectral nulls near $f = n/T$, ($n = 0, 1, 2,$ etc.) and "bumps" within $\pm 1/4MT$ of each null.

V. CONCLUSION

The general analysis of Section II leading to (7) and (13) applies to a wide class of block-coding schemes aside from the one treated here. The simulation/analysis procedures described in Section III can like-

wise be applied to these schemes to find the ρ_q's. Unfortunately, the computer costs involved in accurately estimating the ρ_q's and comparing the resulting formula with simulation data can be quite high.

Aside from computer cost considerations, a strictly theoretical solution to this kind of problem would be more accurate and provide more insight into the correlation factors influencing this kind of random process. Although qualitative explanations can he given for the oscillating hehavior of ρ_q with q (Table II), the approach described here requires and offers little insight into such phenomena.

In strictly practical terms, however, the result of this study provides a spectrum description which is quite accurate and fairly simple to use. For studies involving the passage of block-coded signals through ac-coupled amplifiers, or carrier extraction from signals modulated with block-coded sequences, such descriptions are highly useful.

VI. ACKNOWLEDGMENT

Numerous people have contrihuted to this study with suggestions and assistance. I am particularly indebted to S. O. Rice for his suggestions on comhining theoretical analysis and computer simulation, to Steve Michael for his collaboration in designing the simulation program, and to Diane Vitello for her patience and skill in executing the various computer programs.

REFERENCES

1. F. K. Bowers, "Pulse Code Transmission System," Patent No. 2,957,947, issued Oct. 25, 1960.
2. R. O. Carter, "Low-Disparity Binary Coding System," Elec. Letters, *1*, No. 3 (May 1965), pp. 67–68.
3. J. R. Pierce, unpublished work.
4. C. L. Ruthroff and W. F. Bodtmann, "Adaptive Coding for Coherent Detection of Digital Phase Modulation," B.S.T.J., *53*, No. 3 (March 1974), pp. 449–466.
5. W. F. Bodtmann, unpublished work.
6. S. O. Rice, unpublished work.
7. D. Slepian, unpublished work.
8. J. N. Franklin and J. R. Pierce, "Spectra and Efficiency of Binary Codes without DC," IEEE Trans. on Comm., *COM-20*, No. 6 (Dec. 1972), pp. 1182–1184.
9. B. Gold and C. M. Rader, *Digital Processing of Signals*, New York: McGraw-Hill, 1969, Sec. 6.2.
10. Ref. 9, Secs. 6.4–6.6.
11. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York: McGraw-Hill, 1965, Sec. 8.6.